Package 'TcGSA'

December 22, 2015

Type Package

Title Time-Course Gene Set Analysis

Version 0.10.1

Date 2015-12-22

Depends R (>= 3.0.2), ggplot2 (>= 2.0.0)

Imports lme4 (>= 1.0-4), reshape2, GSA, multtest, gtools, cluster, stringr, splines, gplots, stats, graphics, grDevices, utils

Suggests foreach, parallel, doParallel

Description Implementation of TcGSA, a method for analyzing longitudinal gene-expression data at the gene set level.

License GPL-2 | file LICENSE

BugReports https://github.com/borishejblum/TcGSA/issues

RoxygenNote 5.0.1

NeedsCompilation no

Author Boris P. Hejblum [aut, cre], Damien Chimits [aut]

Maintainer Boris P. Hejblum <bhejblum@hsph.harvard.edu>

Repository CRAN

Date/Publication 2015-12-22 18:12:44

R topics documented:

`TcGSA-package` *Time-course Gene Set Analysis*

Description

This package implements TcGSA, an algorithm to analyze longitudinal gene-expression data at the gene set level.

Details

Package:	TcGSA
Type:	Package
Version:	0.10.1
Date:	2015-12-22
License:	LGPL-3

The main function in this package is `TcGSA.LR` which performs Time-course Gene Set Analysis, and provide nice representations of its results (see `plot.TcGSA` and `plot1GS`).

Author(s)

Boris P. Hejblum

Damien Chimits — Maintainer: Boris P. Hejblum

References

Hejblum BP, Skinner J, Thiebaut R, (2015) Time-Course Gene Set Analysis for Longitudinal Gene Expression Data. *PLoS Computat Biol* 11(6): e1004310. doi: 10.1371/journal.pcbi.1004310

See Also

`GSA`

clustTrend *Cluster the genes dynamics into different dominant trends.*

Description

This function clusters the genes dynamics of one gene sets into different dominant trends. The optimal number of custers is computed thanks to the gap statistics. See clusGap.

Usage

```
clustTrend(tcgs, expr, Subject_ID, TimePoint, threshold = 0.05,
  myproc = "BY", nbsimu_pval = 1e+06, baseline = NULL,
  only.signif = TRUE, group.var = NULL, Group_ID_paired = NULL,
  ref = NULL, group_of_interest = NULL, FUNcluster = NULL,
  clustering_metric = "euclidian", clustering_method = "ward", B = 100,
  max_trends = 4, aggreg.fun = "median", trend.fun = "median",
  methodOptiClust = "firstSEmax", indiv = "genes", verbose = TRUE)

## S3 method for class 'ClusteredTrends'
print(x, ...)

## S3 method for class 'ClusteredTrends'
plot(x, ...)
```

Arguments

tcgs	a **tcgsa** object for clustTrend, or a **ClusteredTrends** object for print.ClusteredTrends and plot.ClusteredTrends.
expr	either a matrix or dataframe of gene expression upon which dynamics are to be calculated, or a list of gene sets estimation of gene expression. In the case of a matrix or dataframe, its dimension are n x p, with the p sample in column and the n genes in row. In the case of a list, its length should correspond to the number of gene sets under scrutiny and each element should be an 3 dimension array of estimated gene expression, such as for the list returned in the 'Estimations' element of TcGSA.LR. See details.
Subject_ID	a factor of length p that is in the same order as the columns of expr (when it is a dataframe) and that contains the patient identifier of each sample.
TimePoint	a numeric vector or a factor of length p that is in the same order as Subject_ID and the columns of expr (when it is a dataframe), and that contains the time points at which gene expression was measured.
threshold	the threshold at which the FDR or the FWER should be controlled.
myproc	a vector of character strings containing the names of the multiple testing procedures for which adjusted p-values are to be computed. This vector should include any of the following: "Bonferroni", "Holm", "Hochberg", "SidakSS", "SidakSD", "BH", "BY", "ABH", "TSBH". See mt.rawp2adjp for details. Default

is "BY", the Benjamini & Yekutieli (2001) step-up FDR-controlling procedure (general dependency structures). In order to control the FWER(in case of an analysis that is more a hypothesis confirmation than an exploration of the expression data), we recommand to use "Holm", the Holm (1979) step-down adjusted p-values for strong control of the FWER.

nbsimu_pval — the number of observations under the null distribution to be generated in order to compute the p-values. Default is 1e+06.

baseline — a character string which is the value of TimePoint that can be used as a baseline. Default is NULL, in which case no timepoint is used as a baseline value for gene expression. Has to be NULL when comparing two treatment groups.

only.signif — logical flag for analysing the trends in only the significant gene sets. If FALSE, all the gene sets from the **gmt** object contained in x are clustered. Default is TRUE.

group.var — in the case of several treatment groups, this is a factor of length p that is in the same order as Timepoint, Subject_ID and the columns of expr. It indicates to which treatment group each sample belongs to. Default is NULL, which means that there is only one treatment group.

Group_ID_paired — a character vector of length p that is in the same order as Timepoint, Subject_ID, group.var and the columns of expr. This argument must not be NULL in the case of a paired analysis, and must be NULL otherwise. Default is NULL.

ref — the group which is used as reference in the case of several treatment groups. Default is NULL, which means that reference is the first group in alphabetical order of the labels of group.var.

group_of_interest — the group of interest, for which dynamics are to be computed in the case of several treatment groups. Default is NULL, which means that group of interest is the second group in alphabetical order of the labels of group.var.

FUNcluster — the clustering function used to agglomerate genes in trends. Default is NULL, in which a hierachical clustering is performed via the function agnes, using the metric clustering_metric and the method clustering_method. See clusGap

clustering_metric — character string specifying the metric to be used for calculating dissimilarities between observations in the hierarchical clustering when FUNcluster is NULL. The currently available options are "euclidean" and "manhattan". Default is "euclidean". See agnes. Also, a "sts" option is available in TcGSA. It implements the 'Short Time Series' distance [Moller-Levet et al., Fuzzy CLustering of short time series and unevenly distributed sampling points, *Advances in Intelligent Data Analysis V*:330-340 Springer, 2003] designed specifically for clustering time series.

clustering_method — character string defining the agglomerative method to be used in the hierarchical clustering when FUNcluster is NULL. The six methods implemented are "average" ([unweighted pair-]group average method, UPGMA), "single" (single linkage), "complete" (complete linkage), "ward" (Ward's method), "weighted" (weighted average linkage). Default is "ward". See agnes.

B	integer specifying the number of Monte Carlo ("bootstrap") samples used to compute the gap statistics. Default is 500. See clusGap.
max_trends	integer specifying the maximum number of different clusters to be tested. Default is 4.
aggreg.fun	a character string such as "mean", "median" or the name of any other defined statistics function that returns a single numeric value. It specifies the function used to aggregate the observations before the clustering. Default is to median. Default is "median".
trend.fun	a character string such as "mean", "median" or the name of any other function that returns a single numeric value. It specifies the function used to calculate the trends of the identified clustered. Default is to median.
methodOptiClust	character string indicating how the "optimal" number of clusters is computed from the gap statistics and their standard deviations. Possible values are "globalmax", "firstmax", "Tibs2001SEmax", "firstSEmax" and "globalSEmax". Default is "firstSEmax". See 'method' in clusGap, Details and *Tibshirani et al., 2001* in References.
indiv	a character string indicating by which unit observations are aggregated (through aggreg.fun) before the clustering. Possible values are "genes" or "patients". Default is "genes".
verbose	logical flag enabling verbose messages to track the computing status of the function. Default is TRUE.
x	an object of class 'ClusteredTrends'.
...	further arguments passed to or from other methods.

Details

If expr is a matrix or a dataframe, then the genes dynamics are clustered on the "original" data. On the other hand, if expr is a list returned in the 'Estimations' element of TcGSA.LR, then the dynamics are computed on the estimations made by the TcGSA.LR function.

This function uses the Gap statistics to determine the optimal number of clusters in the plotted gene set. See clusGap.

Value

An object of class **ClusteredTrends** which is a list with the 4 following components:

- NbClust a vector that contains the optimal number of clusters for each analysed gene sets.
- ClustsMeds a list of the same length as NsClust (the number of analysed gene sets). Each element of the list is a data frame, in which there is as many column as the optimal number of clusters for the corresponding gene setsfor each cluster. Each column of the data frame contains the median trend values for the corresponding cluster.
- GenesPartition a list of the same length as NsClust (the number of analysed gene sets). Each element of the list is a vector which gives the partition of the genes inside the corresponding gene set.
- MaxNbClust an integer storing the maximum number of different clusters tested, as given by the argument 'max_trends'.

Author(s)

Boris P. Hejblum

References

Tibshirani, R., Walther, G. and Hastie, T., 2001, Estimating the number of data clusters via the Gap statistic, *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, **63**, 2: 41–423.

See Also

plot1GS, TcGSA.LR, clusGap

Examples

```
data(data_simu_TcGSA)

tcgsa_sim_1grp <- TcGSA.LR(expr=expr_1grp, gmt=gmt_sim, design=design,
                           subject_name="Patient_ID", time_name="TimePoint",
                           time_func="linear", crossedRandom=FALSE)

CT <- clustTrend(tcgsa_sim_1grp,
    expr=expr_1grp, Subject_ID=design$Subject_ID, TimePoint=design$TimePoint)
CT
plot(CT)

CT$NbClust
CT$NbClust["Gene set 5"]
CT$ClustMeds[["Gene set 4"]]
CT$ClustMeds[["Gene set 5"]]
```

---

data_simu_TcGSA *Simulated Data for TcGSA*

Description

Simulated data for 5 genesets of 50 genes each. Gene expression is simulated at 5 time points for 10 patients.

Usage

```
data(data_simu_TcGSA)
```

Details

In `expr_1grp` all patients belong to the same unique treatment group. The first 2 gene sets are simulated under the null hypothesis. The gene sets 3 and 4 are simulated under the alternative hypothesis that there is a significant homogeneous time trend within the gene set. The gene set 5 is simulated under the alternative hypothesis that there are significant heterogeneous time trends within the gene set.

In `expr_2grp` all patients belong to 2 treatment groups. The 5 first patients belong to the treatment group '`T`', The 5 other patients belong to the treatment group '`C`'. The first 2 gene sets are simulated under the null hypothesis that there is no difference in the time trend between the 2 treatment goups. The gene sets 3 and 4 are simulated under the alternative hypothesis that there are significantly different homogeneous time trends within the gene set between the 2 treatment groups. The gene set 5 is simulated under the alternative hypothesis that there are significantly different heterogeneous time trends between the 2 treatment groups within the gene set.

Value

`expr_1grp` See Details.

`expr_2grp` See Details.

`design` a data frame with 5 variables:

- `Patient_ID`: a factor that contains the patient ID.
- `TimePoint`: a numeric vector or a factor that contains the time points at which gene expression was measured.
- `sample_name`: a character vector with the names of the sample (corresponding to the names of the columns of `expr_1grp` and of `expr_2grp`).
- `group.var`: a factor that indicates to which of the 2 treatment groups each sample belongs to.
- `Group_paired_ID` a random paired identifier for paired couples (one in each of the 2 treatment groups) of patients.

`gmt_sim` a **gmt** object containing the gene sets definition. See `GSA.read.gmt` and GMT definition on www.broadinstitute.org.

Author(s)

Boris P. Hejblum

Source

This is simulated data.

See Also

`TcGSA.LR`

Examples

```
data(data_simu_TcGSA)
summary(expr_1grp)
summary(design)
gmt_sim
```

`multtest.TcGSA` *Computing the P-value of the Likelihood Ratios Applying a Multiple Testing Correction*

Description

This function computes the p-value of the likelihood ratios and apply a multiple testing correction.

Usage

```
multtest.TcGSA(tcgsa, threshold = 0.05, myproc = "BY",
  nbsimu_pval = 1e+06)
```

Arguments

`tcgsa`	a TcGSA object.
`threshold`	the threshold at which the FDR or the FWER should be controlled.
`myproc`	a vector of character strings containing the names of the multiple testing procedures for which adjusted p-values are to be computed. This vector should include any of the following: "Bonferroni", "Holm", "Hochberg", "SidakSS", "SidakSD", "BH", "BY", "ABH", "TSBH" or "none". "none" indicates no adjustement for multiple testing. See `mt.rawp2adjp` for details. Default is "BY", the Benjamini & Yekutieli (2001) step-up FDR-controlling procedure (general dependency structures). In order to control the FWER (in case of an analysis that is more a hypothesis confirmation than an exploration of the expression data), we recommand to use "Holm", the Holm (1979) step-down adjusted p-values for strong control of the FWER.
`nbsimu_pval`	the number of observations under the null distribution to be generated in order to compute the p-values. Default is `1e+06`

Value

`multtest.TcGSA` returns an dataframe with 5 variables. The rows correspond to the gene sets under scrutiny. The 1st column is the likelihood ratios `LR`, the 2nd column is the convergence status of the model under the null hypothesis `CVG_H0`, the 3rd column is the convergence status of the model under the alternative hypothesis `CVG_H1`, the 4th column is the raw p-value of the mixed likelihood ratio test `raw_pval`, the 5th column is the adjusted p-value of the mixed likelihood ratio test `adj_pval`.

Author(s)

Boris P. Hejblum

See Also

TcGSA.LR, mt.rawp2adjp, signifLRT.TcGSA

Examples

```
data(data_simu_TcGSA)

tcgsa_sim_1grp <- TcGSA.LR(expr=expr_1grp, gmt=gmt_sim, design=design,
                           subject_name="Patient_ID", time_name="TimePoint",
                           time_func="linear", crossedRandom=FALSE)

mtt <- multtest.TcGSA(tcgsa_sim_1grp, threshold = 0.05,
                      myproc = "BY", nbsimu_pval = 1000)
mtt
```

---

plot.TcGSA *Plot a Gene Set Trends Heatmap.*

Description

This function plots a gene sets dynamic trends heatmap.

Usage

```
## S3 method for class 'TcGSA'
plot(x, threshold = 0.05, myproc = "BY",
  nbsimu_pval = 1e+06, expr, Subject_ID, TimePoint, baseline = NULL,
  only.signif = TRUE, group.var = NULL, Group_ID_paired = NULL,
  ref = NULL, group_of_interest = NULL, ranking = FALSE,
  FUNcluster = NULL, clustering_metric = "euclidian",
  clustering_method = "ward", B = 500, max_trends = 4,
  aggreg.fun = "median", methodOptiClust = "firstSEmax", indiv = "genes",
  verbose = TRUE, clust_trends = NULL, N_clusters = NULL,
  myclusters = NULL, label.clusters = NULL, prev_rowCL = NULL,
  descript = TRUE, plot = TRUE, color.vec = c("darkred", "#D73027",
  "#FC8D59", "snow", "#91BFDB", "#4575B4", "darkblue"), legend.breaks = NULL,
  label.column = NULL, time_unit = "", cex.label.row = 1,
  cex.label.column = 1, margins = c(5, 25), heatKey.size = 1,
  dendrogram.size = 1, heatmap.height = 1, heatmap.width = 1,
  cex.clusterKey = 1, cex.main = 1, horiz.clusterKey = TRUE,
  main = NULL, subtitle = NULL, ...)
```

Arguments

x	an object of class'TcGSA'.
threshold	the threshold at which the FDR or the FWER should be controlled.
myproc	a vector of character strings containing the names of the multiple testing procedures for which adjusted p-values are to be computed. This vector should include any of the following: "Bonferroni", "Holm", "Hochberg", "SidakSS", "SidakSD", "BH", "BY", "ABH", "TSBH" or "none". "none" indicates no adjustement for multiple testing. See mt.rawp2adjp for details. Default is "BY", the Benjamini & Yekutieli (2001) step-up FDR-controlling procedure (general dependency structures). In order to control the FWER(in case of an analysis that is more a hypothesis confirmation than an exploration of the expression data), we recommand to use "Holm", the Holm (1979) step-down adjusted p-values for strong control of the FWER.
nbsimu_pval	the number of observations under the null distribution to be generated in order to compute the p-values. Default is 1e+06.
expr	either a matrix or dataframe of gene expression upon which dynamics are to be calculated, or a list of gene sets estimation of gene expression. In the case of a matrix or dataframe, its dimension are n x p, with the p sample in column and the n genes in row. In the case of a list, its length should correspond to the number of gene sets under scrutiny and each element should be an 3 dimension array of estimated gene expression, such as for the list returned in the 'Estimations' element of TcGSA.LR. See details.
Subject_ID	a factor of length p that is in the same order as the columns of expr (when it is a dataframe) and that contains the patient identifier of each sample. Ignored if expr is a list of estimations.
TimePoint	a numeric vector or a factor of length p that is in the same order as Subject_ID and the columns of expr (when it is a dataframe), and that contains the time points at which gene expression was measured. Ignored if expr is a list of estimations.
baseline	a character string which is the value of TimePoint used as baseline. See Details.
only.signif	logical flag for plotting only the significant gene sets. If FALSE, all the gene sets from the **gmt** object contained in x are plotted. Default is TRUE.
group.var	in the case of several treatment‘ groups, this is a factor of length p that is in the same order as Timepoint, Subject_ID, sample_name and the columns of expr. It indicates to which treatment group each sample belongs to. Default is NULL, which means that there is only one treatment group. See Details.
Group_ID_paired	a character vector of length p that is in the same order as Timepoint, Subject_ID, sample_name, group.var and the columns of expr. This argument must not be NULL in the case of a paired analysis, and must be NULL otherwise. Default is NULL. See Details.
ref	the group which is used as reference in the case of several treatment groups. Default is NULL, which means that reference is the first group in alphabetical order of the labels of group.var. See Details.

group_of_interest
: the group of interest, for which dynamics are to be computed in the case of several treatment groups. Default is NULL, which means that group of interest is the second group in alphabetical order of the labels of group.var. See Details. group_of_interest here~~

ranking
: a logical flag. If TRUE, the gene set trends are not hierarchicaly classified, but ordered by decreasing Likelihood ratios. Default is FALSE.

FUNcluster
: the clustering function used to agglomerate genes in trends. Default is NULL, in which a hierachical clustering is performed via the function agnes, using the metric clustering_metric and the method clustering_method. See clusGap

clustering_metric
: character string specifying the metric to be used for calculating dissimilarities between observations in the hierarchical clustering when FUNcluster is NULL. The currently available options are "euclidean" and "manhattan". Default is "euclidean". See agnes. Also, a "sts" option is available in TcGSA. It implements the 'Short Time Series' distance [Moller-Levet et al., Fuzzy CLustering of short time series and unevenly distributed sampling points, *Advances in Intelligent Data Analysis V*:330-340 Springer, 2003] designed specifically for clustering time series.

clustering_method
: character string defining the agglomerative method to be used in the hierarchical clustering when FUNcluster is NULL. The six methods implemented are "average" ([unweighted pair-]group average method, UPGMA), "single" (single linkage), "complete" (complete linkage), "ward" (Ward's method), "weighted" (weighted average linkage). Default is "ward". See agnes.

B
: integer specifying the number of Monte Carlo ("bootstrap") samples used to compute the gap statistics. Default is 500. See clusGap.

max_trends
: integer specifying the maximum number of different clusters to be tested. Default is 4.

aggreg.fun
: a character string such as "mean", "median" or the name of any other statistics function defined that returns a single numeric value. It specifies the function used to aggregate the observations before the clustering. Default is to median. Default is "median".

methodOptiClust
: character string indicating how the "optimal"" number of clusters is computed from the gap statistics and their standard deviations. Possible values are "globalmax", "firstmax", "Tibs2001SEmax", "firstSEmax" and "globalSEmax". Default is "firstSEmax". See 'method' in clusGap, Details and *Tibshirani et al., 2001* in References.

indiv
: a character string indicating by which unit observations are aggregated (through aggreg.fun) before the clustering. Possible values are "genes" or "patients". Default is "genes".

verbose
: logical flag enabling verbose messages to track the computing status of the function. Default is TRUE.

clust_trends
: object of class **ClusteredTrends** containing already computed trends for the plotted gene sets. Default is NULL.

N_clusters	an integer that is the number of clusters in which the dynamics should be regrouped. The cutoff of the clustering tree is automatically calculated accordingly. Default is NULL, in which case the dendrogram is not cut and no clusters are identified.
myclusters	a character vector of colors for predefined clusters of the represented genesets, with as many levels as the value of N_clusters. Default is NULL, in which case the clusters are automatically identified and colored via the cutree function and the N_clusters argument only.
label.clusters	if N_clusters is not NULL, a character vector of length N_clusterss. Default is NULL, in which case if N_clusters is not NULL, clusters are simply labelled with numbers.
prev_rowCL	a **hclust** object, such as the one return by the present plotting funstion (see Value) for instance. If not NULL, no clustering is calculated by the present plotting function and this tree is used to represent the gene sets dynamics. Default is NULL.
descript	logical flag indicating that the description of the gene sets should appear after their name on the right side of the plot if TRUE. Default is TRUE. See Details.
plot	logical flag indicating wether the heatmap should be plotted or not. Default is TRUE.
color.vec	a character strings vector used to define the color palette used in the plot. Default is c("#D73027", "#FC8D59","lightyellow", "#91BFDB", "#4575B4").
legend.breaks	a numeric vector indicating the splitting points for coloring. Default is NULL, in which case the break points will be spaced equally and symetrically about 0.
label.column	a vector of character strings with the labels to be displayed for the columns (i.e. the time points). Default is NULL.
time_unit	the time unit to be displayed (such as "Y", "M", "W", "D", "H", etc) next to the values of TimePoint in the columns labels when label.column is NULL. Default is "".
cex.label.row	a numerical value giving the amount by which row labels text should be magnified relative to the default 1.
cex.label.column	a numerical value giving the amount by which column labels text should be magnified relative to the default 1.
margins	numeric vector of length 2 containing the margins (see par(mar= *)) for column and row names, respectively. Default is c(15, 100). See Details.
heatKey.size	the size of the color key for the heatmap fill. Default is 1.
dendrogram.size	the horizontal size of the dendrogram. Default is 1
heatmap.height	the height of the heatmap. Default is 1
heatmap.width	the width of the heatmap. Default is 1
cex.clusterKey	a numerical value giving the amount by which the clusters legend text should be magnified relative to the default 1, when N_clusters is not NULL.
cex.main	a numerical value giving the amount by which title text should be magnified relative to the default 1.

`horiz.clusterKey` a logical flag; if `TRUE`, set the legend for clusters horizontally rather than vertically. Only used if the `N_clusters` argument is not `NULL`. Default is `TRUE`.

`main` a character string for an optionnal title. Default is `NULL`.

`subtitle` a character string for an optionnal subtitle. Default is `NULL`.

`...` other parameters to be passed through to plotting functions.

Details

On the heatmap, each line corresponds to a gene set, and each column to a timepoint.

If `expr` is a matrix or a dataframe, then the "original" data are plotted. On the other hand, if `expr` is a list returned in the `'Estimations'` element of `TcGSA.LR`, then it is those "estimations" made by the `TcGSA.LR` function that are plotted.

If `descript` is `FALSE`, the second element of `margins` can be reduced (for instance use `margins = c(5, 10)`), as there is not so much need for space in order to display only the gene set names, without their description.

If there is a large number of significant gene sets, the hierarchical clustering step repeated for each of them can take a few minutes. To speed things up (especially) when playing with the ploting parameters for having a nice plot, one can run the `clustTrend` function beforehand, and plug its results in the `plot.TcGSA` function via the `clust_trends` argument.

Value

An object of class **hclust** which describes the tree produced by the clustering process. The object is a list with components:

- `merge` an $n-1$ by 2 matrix. Row i of `merge` describes the merging of clusters at step i of the clustering. If an element j in the row is negative, then observation -j was merged at this stage. If j is positive then the merge was with the cluster formed at the (earlier) stage j of the algorithm. Thus negative entries in merge indicate agglomerations of singletons, and positive entries indicate agglomerations of non-singletons.
- `height` a set of $n-1$ real values (non-decreasing for ultrametric trees). The clustering height: that is, the value of the criterion associated with the Ward clustering method.
- `order` a vector giving the permutation of the original observations suitable for plotting, in the sense that a cluster plot using this ordering and matrix merge will not have crossings of the branches.
- `labels` the gene set trends name.
- `call` the call which produced the result clustering:
  `hclust(d = dist(map2heat, method = "euclidean"), method = "ward.D2")`
- `method` "ward.D2", as it is the clustering method that has been used for clustering the gene set trends.
- `dist.method` "euclidean", as it is the distance that has been used for clustering the gene set trends.
- `legend.breaks` a numeric vector giving the splitting points used for coloring the heatmap. If `plot` is `FALSE`, then it is `NULL`.

- `myclusters` a character vector of colors for the dynamic clusters of the represented gene set trends, with as many levels as the value of `N_clusters`. If no dynamic clusters were represented, than this is `NULL`.
- `ddr` a **dendrogram** object with the reordering used for the heatmap. See `heatmap.2`.
- geneset.names character vector with the names of the gene sets used in the heatmap.
- `clust.trends` a **ClusteredTrends** object.
- `clustersExport` a data frame with 2 variables containing the two following variables :
  - `GeneSet`: the gene set trends clustered.
  - `Cluster`: the dynamic cluster they belong to.

  The data frame is order by the variable `Cluster`.
- `data_plotted`: the data matrix represented by the heatmap

Author(s)

Boris P. Hejblum

References

Hejblum BP, Skinner J, Thiebaut R, (2015) Time-Course Gene Set Analysis for Longitudinal Gene Expression Data. *PLoS Computat Biol* 11(6): e1004310. doi: 10.1371/journal.pcbi.1004310

See Also

`heatmap.2`, `TcGSA.LR`, `hclust`

Examples

```
data(data_simu_TcGSA)

tcgsa_sim_1grp <- TcGSA.LR(expr=expr_1grp, gmt=gmt_sim, design=design,
                           subject_name="Patient_ID", time_name="TimePoint",
                           time_func="linear", crossedRandom=FALSE)
summary(tcgsa_sim_1grp)

plot(x=tcgsa_sim_1grp, expr=tcgsa_sim_1grp$Estimations,
    Subject_ID=design$Patient_ID, TimePoint=design$TimePoint,
    baseline=1,
    B=100,
    time_unit="H",
    dendrogram.size=0.4, heatmap.width=0.8, heatmap.height=2, cex.main=0.7
    )

## Not run:
tcgsa_sim_2grp <- TcGSA.LR(expr=expr_2grp, gmt=gmt_sim, design=design,
                           subject_name="Patient_ID", time_name="TimePoint",
                           time_func="linear", crossedRandom=FALSE,
                           group_name="group.var")
summary(tcgsa_sim_2grp)
```

```
plot(x=tcgsa_sim_2grp, expr=expr_2grp,
    Subject_ID=design$Patient_ID, TimePoint=design$TimePoint,
    B=100,
    time_unit="H",
    )

## End(Not run)
```

---

plot1GS *Plotting a Specific Gene Set*

Description

This function can plot different representations of the gene expression in a specific gene set.

Usage

```
plot1GS(expr, gmt, Subject_ID, TimePoint, geneset.name, baseline = NULL,
  group.var = NULL, Group_ID_paired = NULL, ref = NULL,
  group_of_interest = NULL, FUNcluster = NULL,
  clustering_metric = "euclidian", clustering_method = "ward", B = 500,
  max_trends = 4, aggreg.fun = "median", trend.fun = "median",
  methodOptiClust = "firstSEmax", indiv = "genes", verbose = TRUE,
  clustering = TRUE, showTrend = TRUE, smooth = TRUE, precluster = NULL,
  time_unit = "", title = NULL, y.lab = NULL, desc = TRUE,
  lab.cex = 1, axis.cex = 1, main.cex = 1, y.lab.angle = 90,
  x.axis.angle = 45, margins = 1, line.size = 1, y.lim = NULL,
  x.lim = NULL, gg.add = list(theme()), plot = TRUE)
```

Arguments

`expr`	either a matrix or dataframe of gene expression upon which dynamics are to be calculated, or a list of gene sets estimation of gene expression. In the case of a matrix or dataframe, its dimension are n x p, with the p sample in column and the n genes in row. In the case of a list, its length should correspond to the number of gene sets under scrutiny and each element should be an 3 dimension array of estimated gene expression, such as for the list returned in the `'Estimations'` element of `TcGSA.LR`. See details.
`gmt`	a **gmt** object containing the gene sets definition. See `GSA.read.gmt` and definition on www.broadinstitute.org.
`Subject_ID`	a factor of length p that is in the same order as the columns of `expr` (when it is a dataframe) and that contains the patient identifier of each sample.
`TimePoint`	a numeric vector or a factor of length p that is in the same order as `Subject_ID` and the columns of `expr` (when it is a dataframe), and that contains the time points at which gene expression was measured.

`geneset.name` a character string containing the name of the gene set to be plotted, that must appear in the `"geneset.names"` element of `gmt`.

`baseline` a character string which is the value of `TimePoint` that can be used as a baseline. Default is `NULL`, in which case no timepoint is used as a baseline value for gene expression. Has to be `NULL` when comparing two treatment groups.

`group.var` in the case of several treatment groups, this is a factor of length p that is in the same order as `Timepoint`, `Subject_ID` and the columns of `expr`. It indicates to which treatment group each sample belongs to. Default is `NULL`, which means that there is only one treatment group.

`Group_ID_paired`
a character vector of length p that is in the same order as `Timepoint`, `Subject_ID`, `group.var` and the columns of `expr`. This argument must not be `NULL` in the case of a paired analysis, and must be `NULL` otherwise. Default is `NULL`.

`ref` the group which is used as reference in the case of several treatment groups. Default is `NULL`, which means that reference is the first group in alphabetical order of the labels of `group.var`. See Details.

`group_of_interest`
the group of interest, for which dynamics are to be computed in the case of several treatment groups. Default is `NULL`, which means that group of interest is the second group in alphabetical order of the labels of `group.var`.

`FUNcluster` a function which accepts as first argument a matrix `x` and as second argument the number of clusters desired `k`, and which returns a list with a component named `'cluster'` which is a vector of length n = nrow(x) of integers in 1:k, determining the clustering or grouping of the n observations. Default is `NULL`, in which case a hierachical clustering is performed via the function `agnes`, using the metric `clustering_metric` and the method `clustering_method`. See `'FUNcluster'` in `clusGap` and Details.

`clustering_metric`
character string specifying the metric to be used for calculating dissimilarities between observations in the hierarchical clustering when `FUNcluster` is `NULL`. The currently available options are `"euclidean"` and `"manhattan"`. Default is `"euclidean"`. See `agnes`. Also, a `"sts"` option is available in TcGSA. It implements the 'Short Time Series' distance [Moller-Levet et al., Fuzzy CLustering of short time series and unevenly distributed sampling points, *Advances in Intelligent Data Analysis V*:330-340 Springer, 2003] designed specifically for clustering time series.

`clustering_method`
character string defining the agglomerative method to be used in the hierarchical clustering when `FUNcluster` is `NULL`. The six methods implemented are `"average"` ([unweighted pair-]group average method, UPGMA), `"single"` (single linkage), `"complete"` (complete linkage), `"ward"` (Ward's method), `"weighted"` (weighted average linkage). Default is `"ward"`. See `agnes`.

`B` integer specifying the number of Monte Carlo ("bootstrap") samples used to compute the gap statistics. Default is `500`. See `clusGap`.

`max_trends` integer specifying the maximum number of different clusters to be tested. Default is `4`.

`aggreg.fun` a character string such as `"mean"`, `"median"` or the name of any other defined statistics function that returns a single numeric value. It specifies the function used to aggregate the observations before the clustering. Default is to `median`.

`trend.fun` a character string such as `"mean"`, `"median"` or the name of any other function that returns a single numeric value. It specifies the function used to calculate the trends of the identified clustered. Default is to `median`.

`methodOptiClust` character string indicating how the "optimal" number of clusters is computed from the gap statistics and their standard deviations. Possible values are `"globalmax"`, `"firstmax"`, `"Tibs2001SEmax"`, `"firstSEmax"` and `"globalSEmax"`. Default is `"firstSEmax"`. See `'method'` in `clusGap`, Details and *Tibshirani et al., 2001* in References.

`indiv` a character string indicating by which unit observations are aggregated (through `aggreg.fun`) before the clustering. Possible values are `"genes"` or `"patients"`. Default is `"genes"`. See Details.

`verbose` logical flag enabling verbose messages to track the computing status of the function. Default is `TRUE`.

`clustering` logical flag. If `FALSE`, there is no clustering representation; if `TRUE`, the lines are colored according to which cluster they belong to. Default is `TRUE`. See Details.

`showTrend` logical flag. If `TRUE`, a black line is added for each cluster, representing the corresponding `trend.fun`. Default is `TRUE`.

`smooth` logical flag. If `TRUE` and `showTrend` is also `TRUE`, the representation of each cluster `trend.fun` is smoothed using cubic polynoms (see `geom_smooth`. Default is `TRUE`. At the moment, must accept parameter `"na.rm"` (which is automatically set to `TRUE`). This might change in future versions

`precluster` a vector of length p that is in the same order as `Subject_ID`, `TimePoint` and the columns of `expr` (when it is a dataframe), and that contains a prior clustering of the subjects. Default is `NULL`.

`time_unit` the time unit to be displayed (such as `"Y"`, `"M"`, `"W"`, `"D"`, `"H"`, etc) next to the values of `TimePoint` on the x-axis. Default is `""`, in which case the time scale on the x-axis is proportionnal to the time values.

`title` character specifying the title of the plot. If `NULL`, a title is automatically generated, if `""`, no title appears. Default is `NULL`.

`y.lab` character specifying the annotation of the y axis. If `NULL`, an annotation is automatically generated, if `""`, no annotation appears. Default is `NULL`.

`desc` a logical flag. If `TRUE`, a line is added to the title of the plot with the description of the gene set plotted (from the gmt file). Default is `TRUE`.

`lab.cex` a numerical value giving the amount by which lab labels text should be magnified relative to the default `1`.

`axis.cex` a numerical value giving the amount by which axis annotation text should be magnified relative to the default `1`.

`main.cex` a numerical value giving the amount by which title text should be magnified relative to the default `1`.

`y.lab.angle` a numerical value (in [0, 360]) giving the orientation by which y-label text should be turned (anti-clockwise). Default is `90`. See `element_text`.

x.axis.angle	a numerical value (in [0, 360]) giving the orientation by which x-axis annotation text should be turned (anti-clockwise). Default is 45.
margins	a numerical value giving the amount by which the margins should be reduced or increased relative to the default 1.
line.size	a numerical value giving the amount by which the line sizes should be reduced or increased relative to the default 1.
y.lim	a numeric vector of length 2 giving the range of the y-axis. See plot.default.
x.lim	if numeric, will create a continuous scale, if factor or character, will create a discrete scale. Observations not in this range will be dropped. See xlim.
gg.add	A list of instructions to add to the ggplot2 instruction. See +.gg. Default is list(theme()), which adds nothing to the plot.
plot	logical flag. If FALSE, no plot is drawn. Default is TRUE.

Details

If expr is a matrix or a dataframe, then the "original" data are plotted. On the other hand, if expr is a list returned in the 'Estimations' element of TcGSA.LR, then it is those "estimations" made by the TcGSA.LR function that are plotted.

If indiv is 'genes', then each line of the plot is the median of a gene expression over the patients. On the other hand, if indiv is 'patients', then each line of the plot is the median of a patient genes expression in this gene set.

This function uses the Gap statistics to determine the optimal number of clusters in the plotted gene set. See clusGap.

Value

A dataframe the 2 following variables:

- ProbeID which contains the IDs of the probes of the plotted gene set.
- Cluster which to which cluster the probe belongs to.

If clustering is FALSE, then Cluster is NA for all the probes.

Author(s)

Boris P. Hejblum

References

Tibshirani, R., Walther, G. and Hastie, T., 2001, Estimating the number of data clusters via the Gap statistic, *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, **63**, 2: 411–423.

See Also

ggplot, clusGap

Examples

```
data(data_simu_TcGSA)
tcgsa_sim_1grp <- TcGSA.LR(expr=expr_1grp, gmt=gmt_sim, design=design,
                           subject_name="Patient_ID", time_name="TimePoint",
                           time_func="linear", crossedRandom=FALSE)

plot1GS(expr=expr_1grp, TimePoint=design$TimePoint,
       Subject_ID=design$Patient_ID, gmt=gmt_sim,
       geneset.name="Gene set 4",
       indiv="genes", clustering=FALSE,
       time_unit="H",
       lab.cex=0.7)

## Not run:
plot1GS(expr=expr_1grp, TimePoint=design$TimePoint,
       Subject_ID=design$Patient_ID, gmt=gmt_sim,
       geneset.name="Gene set 5",
       indiv="patients", clustering=FALSE, baseline=1,
       time_unit="H",
       lab.cex=0.7)

## End(Not run)
## Not run:
plot1GS(expr=tcgsa_sim_1grp$Estimations, TimePoint=design$TimePoint,
       Subject_ID=design$Patient_ID, gmt=gmt_sim,
       geneset.name="Gene set 5",
       indiv="genes",
       time_unit="H",
       lab.cex=0.7
)

## End(Not run)

## Not run:
library(grDevices)
library(graphics)
colval <- c(hsv(0.56, 0.9, 1),
           hsv(0, 0.27, 1),
           hsv(0.52, 1, 0.5),
           hsv(0, 0.55, 0.97),
           hsv(0.66, 0.15, 1),
           hsv(0, 0.81, 0.55),
           hsv(0.7, 1, 0.7),
           hsv(0.42, 0.33, 1)
)
n <- length(colval);  y <- 1:n
op <- par(mar=rep(1.5,4))
plot(y, axes = FALSE, frame.plot = TRUE,
 xlab = "", ylab = "", pch = 21, cex = 8,
 bg = colval, ylim=c(-1,n+1), xlim=c(-1,n+1),
 main = "Color scale"
```

```
)
par(op)

require(ggplot2)
plot1GS(expr=expr_1grp, TimePoint=design$TimePoint,
       Subject_ID=design$Patient_ID, gmt=gmt_sim,
       geneset.name="Gene set 5",
       indiv="genes",
       time_unit="H",
       title="",
       gg.add=list(scale_color_manual(values=colval),
                   guides(colour = guide_legend(reverse=TRUE))),
       lab.cex=0.7
)

## End(Not run)
```

`plotFit.GS` *Plotting function for exploring the fittness of the mixed modeling used in TcGSA*

Description

This function plots graphs informing on the fit of the mixed modeling of the gene expression performed in TcGSA, for 1 or several gene sets.

Usage

```
plotFit.GS(x, expr, design, subject_name = "Patient_ID",
  time_name = "TimePoint", colnames_ID, plot_type = c("Fit",
  "Residuals Obs", "Residuals Est", "Histogram Obs"), GeneSetsList,
  color = c("genes", "time", "subjects"), marginal_hist = TRUE,
  gg.add = list(theme()))
```

Arguments

`x`	a **tcgsa** object for `clustTrend`, or a **ClusteredTrends** object for `print.ClusteredTrends` and `plot.ClusteredTrends`.
`expr`	a matrix or dataframe of gene expression. Its dimension are nxp, with the p samples in column and the n genes in row.
`design`	a matrix or dataframe containing the experimental variables that used in the model, namely `subject_name`, `time_name`, and `covariates_fixed` and `time_covariates` if applicable. Its dimension are pxm and its row are is in the same order as the columns of `expr`.
`subject_name`	the name of the factor variable from `design` that contains the information on the repetition units used in the mixed model, such as the patient identifiers for instance. Default is `'Patient_ID'`. See Details.

time_name	the name of a numeric variable from design that contains the information on the time replicates (the time points at which gene expression was measured). Default is 'TimePoint'. See Details.
colnames_ID	the name of the variable from design that contains the columnames of the expr expression data matrix. See Details.
plot_type	a character string indicating the type of plot to be drawn. The options are 'Fit', 'Residuals Obs', 'Residuals Est' or 'Histogram Obs'.
GeneSetsList	a character string containing the names of the gene set whose fit is being checked. If several gene sets are being checked, can be a character list or vector of the names of those gene sets.
color	a character string indicating which color scale should be used. One of the 3 : 'genes', 'time', 'subjects', otherwise, no coloring is used.
marginal_hist	a logical flag indicating wether marginal histograms should be drawn. Only used for 'Fit' plot type. Default is 'TRUE'
gg.add	A list of instructions to add to the ggplot2 instruction. See +.gg. Default is list(theme()), which adds nothing to the plot.

Author(s)

Boris P. Hejblum

References

Hejblum BP, Skinner J, Thiebaut R, (2015) Time-Course Gene Set Analysis for Longitudinal Gene Expression Data. *PLoS Computat Biol* 11(6): e1004310. doi: 10.1371/journal.pcbi.1004310

See Also

plot1GS, plotSelect.GS

Examples

```
data(data_simu_TcGSA)

tcgsa_sim_1grp <- TcGSA.LR(expr=expr_1grp, gmt=gmt_sim, design=design,
                           subject_name="Patient_ID", time_name="TimePoint",
                           time_func="linear", crossedRandom=FALSE)
plotFit.GS(x=tcgsa_sim_1grp, expr=expr_1grp, design=design,
 subject_name="Patient_ID", time_name="TimePoint",
 colnames_ID="Sample_name",
 plot_type="Residuals Obs",
 GeneSetsList=c("Gene set 1", "Gene set 2", "Gene set 3",
                "Gene set 4", "Gene set 5"),
 color="genes", gg.add=list(guides(color=FALSE))
)

## Not run:
plotFit.GS(x=tcgsa_sim_1grp, expr=expr_1grp, design=design,
          subject_name="Patient_ID", time_name="TimePoint",
```

```
            colnames_ID="Sample_name",
            plot_type="Histogram Obs",
            GeneSetsList=c("Gene set 1", "Gene set 5"),
            color="genes", gg.add=list(guides(fill=FALSE))
            )

plotFit.GS(x=tcgsa_sim_1grp, expr=expr_1grp, design=design,
            subject_name="Patient_ID", time_name="TimePoint",
            colnames_ID="Sample_name",
            plot_type="Histogram Obs",
            GeneSetsList=c("Gene set 1", "Gene set 2", "Gene set 3",
                  "Gene set 4", "Gene set 5"),
            color="genes")

## End(Not run)
```

---

plotPat.1GS *Plotting a Specific Gene Set Stratifying on Patients*

Description

This function can plot different representations of the gene expression in a specific gene set, stratified on all subjects.

Usage

```
plotPat.1GS(expr, gmt, Subject_ID, TimePoint, geneset.name, baseline = NULL,
  group.var = NULL, Group_ID_paired = NULL, ref = NULL,
  group_of_interest = NULL, FUNcluster = NULL,
  clustering_metric = "euclidian", clustering_method = "ward", B = 500,
  max_trends = 4, aggreg.fun = "median", trend.fun = "median",
  methodOptiClust = "firstSEmax", verbose = TRUE, clustering = TRUE,
  time_unit = "", title = NULL, y.lab = NULL, desc = TRUE,
  lab.cex = 1, axis.cex = 1, main.cex = 1, y.lab.angle = 90,
  x.axis.angle = 45, y.lim = NULL, x.lim = NULL, gg.add = list(theme()))
```

Arguments

expr	either a matrix or dataframe of gene expression upon which dynamics are to be calculated, or a list of gene sets estimation of gene expression. In the case of a matrix or dataframe, its dimension are n x p, with the p sample in column and the n genes in row. In the case of a list, its length should correspond to the number of gene sets under scrutiny and each element should be an 3 dimension array of estimated gene expression, such as for the list returned in the 'Estimations' element of TcGSA.LR. See details.
gmt	a **gmt** object containing the gene sets definition. See GSA.read.gmt and definition on www.broadinstitute.org.

`Subject_ID` a factor of length p that is in the same order as the columns of `expr` (when it is a dataframe) and that contains the patient identifier of each sample.

`TimePoint` a numeric vector or a factor of length p that is in the same order as `TimePoint` and the columns of `expr` (when it is a dataframe), and that contains the time points at which gene expression was measured.

`geneset.name` a character string containing the name of the gene set to be plotted, that must appear in the `"geneset.names"` element of `gmt`.

`baseline` a character string which is the value of `TimePoint` that can be used as a baseline. Default is `NULL`, in which case no timepoint is used as a baseline value for gene expression. Has to be `NULL` when comparing two treatment groups.

`group.var` in the case of several treatment groups, this is a factor of length p that is in the same order as `Timepoint`, `Subject_ID` and the columns of `expr`. It indicates to which treatment group each sample belongs to. Default is `NULL`, which means that there is only one treatment group. See Details.

`Group_ID_paired`
a character vector of length p that is in the same order as `Timepoint`, `Subject_ID`, `group.var` and the columns of `expr`. This argument must not be `NULL` in the case of a paired analysis, and must be `NULL` otherwise. Default is `NULL`.

`ref` the group which is used as reference in the case of several treatment groups. Default is `NULL`, which means that reference is the first group in alphabetical order of the labels of `group.var`. See Details.

`group_of_interest`
the group of interest, for which dynamics are to be computed in the case of several treatment groups. Default is `NULL`, which means that group of interest is the second group in alphabetical order of the labels of `group.var`.

`FUNcluster` a function which accepts as first argument a matrix `x` and as second argument the number of clusters desired `k`, and which returns a list with a component named `'cluster'` which is a vector of length n = nrow(x) of integers in 1:k, determining the clustering or grouping of the n observations. Default is `NULL`, in which case a hierachical clustering is performed via the function agnes, using the metric `clustering_metric` and the method `clustering_method`. See `'FUNcluster'` in clusGap and Details.

`clustering_metric`
character string specifying the metric to be used for calculating dissimilarities between observations in the hierarchical clustering when `FUNcluster` is `NULL`. The currently available options are `"euclidean"` and `"manhattan"`. Default is `"euclidean"`. See agnes. Also, a `"sts"` option is available in TcGSA. It implements the 'Short Time Series' distance [Moller-Levet et al., Fuzzy CLustering of short time series and unevenly distributed sampling points, *Advances in Intelligent Data Analysis V*:330-340 Springer, 2003] designed specifically for clustering time series.

`clustering_method`
character string defining the agglomerative method to be used in the hierarchical clustering when `FUNcluster` is `NULL`. The six methods implemented are `"average"` ([unweighted pair-]group average method, UPGMA), `"single"` (single linkage), `"complete"` (complete linkage), `"ward"` (Ward's method), `"weighted"` (weighted average linkage). Default is `"ward"`. See agnes.

B	integer specifying the number of Monte Carlo ("bootstrap") samples used to compute the gap statistics. Default is 500. See clusGap.
max_trends	integer specifying the maximum number of different clusters to be tested. Default is 4.
aggreg.fun	a character string such as "mean", "median" or the name of any other defined statistics function that returns a single numeric value. It specifies the function used to aggregate the observations before the clustering. Default is to median.
trend.fun	a character string such as "mean", "median" or the name of any other function that returns a single numeric value. It specifies the function used to calculate the trends of the identified clustered. Default is to median.
methodOptiClust	character string indicating how the "optimal" number of clusters is computed from the gap statistics and their standard deviations. Possible values are "globalmax", "firstmax", "Tibs2001SEmax", "firstSEmax" and "globalSEmax". Default is "firstSEmax". See 'method' in clusGap, Details and *Tibshirani et al., 2001* in References.
verbose	logical flag enabling verbose messages to track the computing status of the function. Default is TRUE.
clustering	logical flag. If FALSE, there is no clustering representation; if TRUE, the lines are colored according to which cluster they belong to. Default is TRUE. See Details.
time_unit	the time unit to be displayed (such as "Y", "M", "W", "D", "H", etc) next to the values of TimePoint on the x-axis. Default is "".
title	character specifying the title of the plot. If NULL, a title is automatically generated, if "", no title appears. Default is NULL.
y.lab	character specifying the annotation of the y axis. If NULL, an annotation is automatically generated, if "", no annotation appears. Default is NULL.
desc	a logical flag. If TRUE, a line is added to the title of the plot with the description of the gene set plotted (from the gmt file). Default is TRUE.
lab.cex	a numerical value giving the amount by which lab labels text should be magnified relative to the default 1.
axis.cex	a numerical value giving the amount by which axis annotation text should be magnified relative to the default 1.
main.cex	a numerical value giving the amount by which title text should be magnified relative to the default 1.
y.lab.angle	a numerical value (in [0, 360]) giving the orientation by which y-label text should be turned (anti-clockwise). Default is 90. See element_text.
x.axis.angle	a numerical value (in [0, 360]) giving the orientation by which x-axis annotation text should be turned (anti-clockwise). Default is 45.
y.lim	a numeric vector of length 2 giving the range of the y-axis. See plot.default.
x.lim	if numeric, will create a continuous scale, if factor or character, will create a discrete scale. Observations not in this range will be dropped. See xlim.
gg.add	A list of instructions to add to the ggplot2 instruction. See +.gg. Default is list(theme()), which adds nothing to the plot.

Details

If expr is a matrix or a dataframe, then the "original" data are plotted. On the other hand, if expr is a list returned in the 'Estimations' element of TcGSA.LR, then it is those "estimations" made by the TcGSA.LR function that are plotted.

If indiv is 'genes', then each line of the plot is the median of a gene expression over the patients. On the other hand, if indiv is 'patients', then each line of the plot is the median of a patient genes expression in this gene set.

This function uses the Gap statistics to determine the optimal number of clusters in the plotted gene set. See clusGap.

Value

A dataframe the 2 following variables:

- ProbeID which contains the IDs of the probes of the plotted gene set.
- Cluster which to which cluster the probe belongs to.

If clustering is FALSE, then Cluster is NA for all the probes.

Author(s)

Boris P. Hejblum

References

Tibshirani, R., Walther, G. and Hastie, T., 2001, Estimating the number of data clusters via the Gap statistic, *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, **63**, 2: 411–423.

See Also

ggplot, clusGap

Examples

```
data(data_simu_TcGSA)
tcgsa_sim_1grp <- TcGSA.LR(expr=expr_1grp, gmt=gmt_sim, design=design,
                           subject_name="Patient_ID", time_name="TimePoint",
                           time_func="linear", crossedRandom=FALSE)

plotPat.1GS(expr=expr_1grp, TimePoint=design$TimePoint,
       Subject_ID=design$Patient_ID, gmt=gmt_sim,
       geneset.name="Gene set 4",
       clustering=FALSE,
       time_unit="H",
       lab.cex=0.7)

## Not run:
plotPat.1GS(expr=expr_1grp, TimePoint=design$TimePoint,
```

```
       Subject_ID=design$Patient_ID, gmt=gmt_sim,
       geneset.name="Gene set 4",
       clustering=FALSE, baseline=1,
       time_unit="H",
       lab.cex=0.7)

## End(Not run)

## Not run:
colval <- c(hsv(0.56, 0.9, 1),
           hsv(0, 0.27, 1),
           hsv(0.52, 1, 0.5),
           hsv(0, 0.55, 0.97),
           hsv(0.66, 0.15, 1),
           hsv(0, 0.81, 0.55),
           hsv(0.7, 1, 0.7),
           hsv(0.42, 0.33, 1)
)
n <- length(colval);  y <- 1:n
op <- par(mar=rep(1.5,4))
plot(y, axes = FALSE, frame.plot = TRUE,
 xlab = "", ylab = "", pch = 21, cex = 8,
 bg = colval, ylim=c(-1,n+1), xlim=c(-1,n+1),
 main = "Color scale"
)
par(op)

plotPat.1GS(expr=expr_1grp, TimePoint=design$TimePoint,
       Subject_ID=design$Patient_ID, gmt=gmt_sim,
       geneset.name="Gene set 5",
       time_unit="H",
       title="",
       gg.add=list(scale_color_manual(values=colval)),
       lab.cex=0.7
)

## End(Not run)

## Not run:
plotPat.1GS(expr=tcgsa_sim_1grp$Estimations, TimePoint=design$TimePoint,
       Subject_ID=design$Patient_ID, gmt=gmt_sim,
       geneset.name="Gene set 3",
       time_unit="H",
       lab.cex=0.7
)

## End(Not run)
```

---

plotPat.TcGSA *Plot a Gene Set Trends Heatmap for each Patient.*

Description

This function plots a series of gene sets dynamic trends heatmaps. One heatmap is drawned for each patient. NOT IMPLEMENTED YET (TODO)

Usage

```
plotPat.TcGSA(x, threshold = 0.05, myproc = "BY", nbsimu_pval = 1e+06,
  expr, Subject_ID, TimePoint, baseline = NULL, only.signif = TRUE,
  group.var = NULL, Group_ID_paired = NULL, ref = NULL,
  group_of_interest = NULL, FUNcluster = NULL,
  clustering_metric = "euclidian", clustering_method = "ward", B = 500,
  max_trends = 4, aggreg.fun = "median", methodOptiClust = "firstSEmax",
  verbose = TRUE, clust_trends = NULL, N_clusters = NULL,
  myclusters = NULL, label.clusters = NULL, prev_rowCL = NULL,
  descript = TRUE, plotAll = TRUE, color.vec = c("darkred", "#D73027",
  "#FC8D59", "snow", "#91BFDB", "#4575B4", "darkblue"), legend.breaks = NULL,
  label.column = NULL, time_unit = "", cex.label.row = 1,
  cex.label.column = 1, margins = c(5, 25), heatKey.size = 1,
  dendrogram.size = 1, heatmap.height = 1, heatmap.width = 1,
  cex.clusterKey = 1, cex.main = 1, horiz.clusterKey = TRUE,
  main = NULL, subtitle = NULL, ...)
```

Arguments

x	a **tcgsa** object.
threshold	the threshold at which the FDR or the FWER should be controlled.
myproc	a vector of character strings containing the names of the multiple testing procedures for which adjusted p-values are to be computed. This vector should include any of the following: "Bonferroni", "Holm", "Hochberg", "SidakSS", "SidakSD", "BH", "BY", "ABH", "TSBH" or "none". "none" indicates no adjustement for multiple testing. See mt.rawp2adjp for details. Default is "BY", the Benjamini & Yekutieli (2001) step-up FDR-controlling procedure (general dependency structures). In order to control the FWER(in case of an analysis that is more a hypothesis confirmation than an exploration of the expression data), we recommand to use "Holm", the Holm (1979) step-down adjusted p-values for strong control of the FWER.
nbsimu_pval	the number of observations under the null distribution to be generated in order to compute the p-values. Default is 1e+06.
expr	either a matrix or dataframe of gene expression upon which dynamics are to be calculated, or a list of gene sets estimation of gene expression. In the case of a matrix or dataframe, its dimension are n x p, with the p sample in column and the n genes in row. In the case of a list, its length should correspond to the number of gene sets under scrutiny and each element should be an 3 dimension array of estimated gene expression, such as for the list returned in the 'Estimations' element of TcGSA.LR. See Details.
Subject_ID	a factor of length p that is in the same order as the columns of expr (when it is a dataframe) and that contains the patient identifier of each sample.

`TimePoint` a numeric vector or a factor of length p that is in the same order as `Subject_ID` and the columns of `expr` (when it is a dataframe), and that contains the time points at which gene expression was measured.

`baseline` a character string which is the value of `TimePoint` used as baseline.

`only.signif` logical flag for plotting only the significant gene sets. If `FALSE`, all the gene sets from the **gmt** object contained in `x` are plotted. Default is `TRUE`.

`group.var` in the case of several treatment groups, this is a factor of length p that is in the same order as `Timepoint`, `Subject_ID`, `sample_name` and the columns of `expr`. It indicates to which treatment group each sample belongs to. Default is `NULL`, which means that there is only one treatment group. See Details.

`Group_ID_paired` a character vector of length p that is in the same order as `Timepoint`, `Subject_ID`, `sample_name`, `group.var` and the columns of `expr`. This argument must not be `NULL` in the case of a paired analysis, and must be `NULL` otherwise. Default is `NULL`.

`ref` the group which is used as reference in the case of several treatment groups. Default is `NULL`, which means that reference is the first group in alphabetical order of the labels of `group.var`. See Details.

`group_of_interest` the group of interest, for which dynamics are to be computed in the case of several treatment groups. Default is `NULL`, which means that group of interest is the second group in alphabetical order of the labels of `group.var`.

`FUNcluster` the clustering function used to agglomerate genes in trends. Default is `NULL`, in which a hierachical clustering is performed via the function `agnes`, using the metric `clustering_metric` and the method `clustering_method`. See `clusGap`

`clustering_metric` character string specifying the metric to be used for calculating dissimilarities between observations in the hierarchical clustering when `FUNcluster` is `NULL`. The currently available options are `"euclidean"` and `"manhattan"`. Default is `"euclidean"`. See `agnes`. Also, a `"sts"` option is available in TcGSA. It implements the 'Short Time Series' distance [Moller-Levet et al., Fuzzy CLustering of short time series and unevenly distributed sampling points, *Advances in Intelligent Data Analysis V*:330-340 Springer, 2003] designed specifically for clustering time series.

`clustering_method` character string defining the agglomerative method to be used in the hierarchical clustering when `FUNcluster` is `NULL`. The six methods implemented are `"average"` ([unweighted pair-]group average method, UPGMA), `"single"` (single linkage), `"complete"` (complete linkage), `"ward"` (Ward's method), `"weighted"` (weighted average linkage). Default is `"ward"`. See `agnes`.

`B` integer specifying the number of Monte Carlo ("bootstrap") samples used to compute the gap statistics. Default is `500`. See `clusGap`.

`max_trends` integer specifying the maximum number of different clusters to be tested. Default is `4`.

`aggreg.fun` a character string such as `"mean"`, `"median"` or the name of any other statistics function defined that returns a single numeric value. It specifies the function

used to aggregate the observations before the clustering. Default is to median. Default is "median".

methodOptiClust
: character string indicating how the "optimal"" number of clusters is computed from the gap statistics and their standard deviations. Possible values are "globalmax", "firstmax", "Tibs2001SEmax", "firstSEmax" and "globalSEmax". Default is "firstSEmax". See 'method' in clusGap, Details and *Tibshirani et al., 2001* in References.

verbose
: logical flag enabling verbose messages to track the computing status of the function. Default is TRUE.

clust_trends
: object of class **ClusteredTrends** containing already computed trends for the plotted gene sets. Default is NULL.

N_clusters
: an integer that is the number of clusters in which the dynamics should be regrouped. The cutoff of the clustering tree is automatically calculated accordingly. Default is NULL, in which case the dendrogram is not cut and no clusters are identified.

myclusters
: a character vector of colors for predefined clusters of the represented genesets, with as many levels as the value of N_clusters. Default is NULL, in which case the clusters are automatically identified and colored via the cutree function and the N_clusters argument only.

label.clusters
: if N_clusters is not NULL, a character vector of length N_clusterss. Default is NULL, in which case if N_clusters is not NULL, clusters are simply labelled with numbers.

prev_rowCL
: a **hclust** object, such as the one return by the present plotting funstion (see Value) for instance. If not NULL, no clustering is calculated by the present plotting function and this tree is used to represent the gene sets dynamics. Default is NULL.

descript
: logical flag indicating that the description of the gene sets should appear after their name on the right side of the plot if TRUE. Default is TRUE. See Details.

plotAll
: logical flag indicating wether a first heatmap with the median over all the patients should be plotted, or not. Default is TRUE.

color.vec
: a character strings vector used to define the color palette used in the plot. Default is c("#D73027", "#FC8D59","lightyellow", "#91BFDB", "#4575B4").

legend.breaks
: a numeric vector indicating the splitting points for coloring. Default is NULL, in which case the break points will be spaced equally and symetrically about 0.

label.column
: a vector of character strings with the labels to be displayed for the columns (i.e. the time points). Default is NULL.

time_unit
: the time unit to be displayed (such as "Y", "M", "W", "D", "H", etc) next to the values of TimePoint in the columns labels when label.column is NULL. Default is "".

cex.label.row
: a numerical value giving the amount by which row labels text should be magnified relative to the default 1.

cex.label.column
: a numerical value giving the amount by which column labels text should be magnified relative to the default 1.

margins	numeric vector of length 2 containing the margins (see par(mar= *)) for column and row names, respectively. Default is c(15, 100). See Details.
heatKey.size	the size of the color key for the heatmap fill. Default is 1.
dendrogram.size	the horizontal size of the dendrogram. Default is 1
heatmap.height	the height of the heatmap. Default is 1
heatmap.width	the width of the heatmap. Default is 1
cex.clusterKey	a numerical value giving the amount by which the clusters legend text should be magnified relative to the default 1, when N_clusters is not NULL.
cex.main	a numerical value giving the amount by which title text should be magnified relative to the default 1.
horiz.clusterKey	a logical flag; if TRUE, set the legend for clusters horizontally rather than vertically. Only used if the N_clusters argument is not NULL. Default is TRUE.
main	a character string for an optionnal title. Default is NULL.
subtitle	a character string for an optionnal subtitle. Default is NULL.
...	other parameters to be passed through to plotting functions.

Details

On the heatmap, each line corresponds to a gene set, and each column to a timepoint.

First a heatmap is computed on all the patients (see plot.TcGSA and clustTrend) to define the clustering. Then, the clustering and coloring thus defined on all the patients are consistently used in the separate heatmaps that are plotted by patient.

If expr is a matrix or a dataframe, then the "original" data are plotted. On the other hand, if expr is a list returned in the 'Estimations' element of TcGSA.LR, then it is those "estimations" made by the TcGSA.LR function that are plotted.

If descript is FALSE, the second element of margins can be reduced (for instance use margins = c(5, 10)), as there is not so much need for space in order to display only the gene set names, without their description.

The median shown in the heatmap uses the respectively standardized (reduced and centered) expression of the genes over the patients.

Value

An object of class **hclust** which describes the tree produced by the clustering process. The object is a list with components:

- merge an $n-1$ by 2 matrix. Row i of merge describes the merging of clusters at step i of the clustering. If an element j in the row is negative, then observation -j was merged at this stage. If j is positive then the merge was with the cluster formed at the (earlier) stage j of the algorithm. Thus negative entries in merge indicate agglomerations of singletons, and positive entries indicate agglomerations of non-singletons.
- height a set of $n-1$ real values (non-decreasing for ultrametric trees). The clustering height: that is, the value of the criterion associated with the Ward clustering method.

- order a vector giving the permutation of the original observations suitable for plotting, in the sense that a cluster plot using this ordering and matrix merge will not have crossings of the branches.
- labels the gene sets name.
- call the call which produced the result clustering:
  `hclust(d = dist(map2heat, method = "euclidean"), method = "ward.D2")`
- method "ward.D2", as it is the clustering method that has been used for clustering the gene set trends.
- dist.method "euclidean", as it is the distance that has been used for clustering the gene set trends.
- legend.breaks a numeric vector giving the splitting points used for coloring the heatmap. If `plot` is `FALSE`, then it is `NULL`.
- myclusters a character vector of colors for clusters of the represented genesets, with as many levels as the value of `N_clusters`. If no clusters were represented, than this is `NULL`.
- ddr a **dendrogram** object with the reordering used for the heatmap. See `heatmap.2`.
- clustersExport a data frame with 2 variables containing the two following variables :
  - `GeneSet`: the gene sets clustered.
  - `Cluster`: the cluster they belong to.

  The data frame is order by the variable `Cluster`.

Author(s)

Boris P. Hejblum

References

Hejblum BP, Skinner J, Thiebaut R, (2015) Time-Course Gene Set Analysis for Longitudinal Gene Expression Data. *PLoS Computat Biol* 11(6): e1004310. doi: 10.1371/journal.pcbi.1004310

See Also

`plot.TcGSA`, `heatmap.2`, `TcGSA.LR`, `hclust`

Examples

```
data(data_simu_TcGSA)

tcgsa_sim_1grp <- TcGSA.LR(expr=expr_1grp, gmt=gmt_sim, design=design,
                           subject_name="Patient_ID", time_name="TimePoint",
                           time_func="linear", crossedRandom=FALSE)

plotPat.TcGSA(x=tcgsa_sim_1grp, expr=expr_1grp,
    Subject_ID=design$Patient_ID, TimePoint=design$TimePoint,
    B=100,
    time_unit="H"
    )
```

```
plotPat.TcGSA(x=tcgsa_sim_1grp, expr=tcgsa_sim_1grp$Estimations,
    Subject_ID=design$Patient_ID, TimePoint=design$TimePoint,
    baseline=1,
    B=100,
    time_unit="H"
    )
```

---

plotSelect.GS *Plotting (several) Selected Gene Set(s) in some Subjects*

Description

This function can plot different representations of the gene expression in selected gene sets, among a subset of selected subjects.

Usage

```
plotSelect.GS(expr, gmt, Subject_ID, TimePoint, geneset.names.select,
  Subject_ID.select, display = "one subject per page", baseline = NULL,
  group.var = NULL, Group_ID_paired = NULL, ref = NULL,
  group_of_interest = NULL, FUNcluster = NULL,
  clustering_metric = "euclidian", clustering_method = "ward", B = 500,
  max_trends = 4, aggreg.fun = "median", trend.fun = "median",
  methodOptiClust = "firstSEmax", verbose = TRUE, clustering = TRUE,
  time_unit = "", title = NULL, y.lab = NULL, desc = TRUE,
  lab.cex = 1, axis.cex = 1, main.cex = 1, y.lab.angle = 90,
  x.axis.angle = 45, y.lim = NULL, x.lim = NULL, gg.add = list(theme()))
```

Arguments

expr	either a matrix or dataframe of gene expression upon which dynamics are to be calculated, or a list of gene sets estimation of gene expression. In the case of a matrix or dataframe, its dimension are n x p, with the p sample in column and the n genes in row. In the case of a list, its length should correspond to the number of gene sets under scrutiny and each element should be an 3 dimension array of estimated gene expression, such as for the list returned in the 'Estimations' element of TcGSA.LR. See Details.
gmt	a **gmt** object containing the gene sets definition. See GSA.read.gmt and definition on www.broadinstitute.org.
Subject_ID	a factor of length p that is in the same order as the columns of expr (when it is a dataframe) and that contains the patient identifier of each sample.
TimePoint	a numeric vector or a factor of length p that is in the same order as TimePoint and the columns of expr (when it is a dataframe), and that contains the time points at which gene expression was measured.

`geneset.names.select`
: a character vector containing the names of the gene sets to be plotted, that must appear in the `"geneset.names"` element of `gmt`.

`Subject_ID.select`
: a character vector containing the names of the subjects to be plotted, that must appear in the `Subject_ID` vector.

`display`
: How to display the resulting graphs. One of the following : `"one GS per page"`, `"one subject per page"`, `"median over selected patients"`. Default is `"one subject per page"`.

`baseline`
: a character string which is the value of `TimePoint` that can be used as a baseline. Default is `NULL`, in which case no timepoint is used as a baseline value for gene expression. Has to be `NULL` when comparing two treatment groups.

`group.var`
: in the case of several treatment groups, this is a factor of length p that is in the same order as `Timepoint`, `Subject_ID` and the columns of `expr`. It indicates to which treatment group each sample belongs to. Default is `NULL`, which means that there is only one treatment group.

`Group_ID_paired`
: a character vector of length p that is in the same order as `Timepoint`, `Subject_ID`, `group.var` and the columns of `expr`. This argument must not be `NULL` in the case of a paired analysis, and must be `NULL` otherwise. Default is `NULL`.

`ref`
: the group which is used as reference in the case of several treatment groups. Default is `NULL`, which means that reference is the first group in alphabetical order of the labels of `group.var`. See Details.

`group_of_interest`
: the group of interest, for which dynamics are to be computed in the case of several treatment groups. Default is `NULL`, which means that group of interest is the second group in alphabetical order of the labels of `group.var`.

`FUNcluster`
: a function which accepts as first argument a matrix `x` and as second argument the number of clusters desired `k`, and which returns a list with a component named `'cluster'` which is a vector of length n = nrow(x) of integers in 1:k, determining the clustering or grouping of the n observations. Default is `NULL`, in which case a hierachical clustering is performed via the function `agnes`, using the metric `clustering_metric` and the method `clustering_method`. See `'FUNcluster'` in `clusGap` and Details.

`clustering_metric`
: character string specifying the metric to be used for calculating dissimilarities between observations in the hierarchical clustering when `FUNcluster` is `NULL`. The currently available options are `"euclidean"` and `"manhattan"`. Default is `"euclidean"`. See `agnes`. Also, a `"sts"` option is available in TcGSA. It implements the 'Short Time Series' distance [Moller-Levet et al., Fuzzy CLustering of short time series and unevenly distributed sampling points, *Advances in Intelligent Data Analysis V*:330-340 Springer, 2003] designed specifically for clustering time series.

`clustering_method`
: character string defining the agglomerative method to be used in the hierarchical clustering when `FUNcluster` is `NULL`. The six methods implemented are

	"average" ([unweighted pair-]group average method, UPGMA), "single" (single linkage), "complete" (complete linkage), "ward" (Ward's method), "weighted" (weighted average linkage). Default is "ward". See agnes.
B	integer specifying the number of Monte Carlo ("bootstrap") samples used to compute the gap statistics. Default is 500. See clusGap.
max_trends	integer specifying the maximum number of different clusters to be tested. Default is 4.
aggreg.fun	a character string such as "mean", "median" or the name of any other defined statistics function that returns a single numeric value. It specifies the function used to aggregate the observations before the clustering. Default is to median.
trend.fun	a character string such as "mean", "median" or the name of any other function that returns a single numeric value. It specifies the function used to calculate the trends of the identified clustered. Default is to median.
methodOptiClust	character string indicating how the "optimal" number of clusters is computed from the gap statistics and their standard deviations. Possible values are "globalmax", "firstmax", "Tibs2001SEmax", "firstSEmax" and "globalSEmax". Default is "firstSEmax". See 'method' in clusGap, Details and *Tibshirani et al., 2001* in References.
verbose	logical flag enabling verbose messages to track the computing status of the function. Default is TRUE.
clustering	logical flag. If FALSE, there is no clustering representation; if TRUE, the lines are colored according to which cluster they belong to. Default is TRUE. See Details.
time_unit	the time unit to be displayed (such as "Y", "M", "W", "D", "H", etc) next to the values of TimePoint on the x-axis. Default is "".
title	character specifying the title of the plot. If NULL, a title is automatically generated, if "", no title appears. Default is NULL.
y.lab	character specifying the annotation of the y axis. If NULL, an annotation is automatically generated, if "", no annotation appears. Default is NULL.
desc	a logical flag. If TRUE, a line is added to the title of the plot with the description of the gene set plotted (from the gmt file). Default is TRUE.
lab.cex	a numerical value giving the amount by which lab labels text should be magnified relative to the default 1.
axis.cex	a numerical value giving the amount by which axis annotation text should be magnified relative to the default 1.
main.cex	a numerical value giving the amount by which title text should be magnified relative to the default 1.
y.lab.angle	a numerical value (in [0, 360]) giving the orientation by which y-label text should be turned (anti-clockwise). Default is 90. See element_text.
x.axis.angle	a numerical value (in [0, 360]) giving the orientation by which x-axis annotation text should be turned (anti-clockwise). Default is 45.
y.lim	a numeric vector of length 2 giving the range of the y-axis. See plot.default.
x.lim	if numeric, will create a continuous scale, if factor or character, will create a discrete scale. Observations not in this range will be dropped. See xlim.

gg.add — A list of instructions to add to the ggplot2 instruction. See +.gg. Default is list(theme()), which adds nothing to the plot.

Details

If expr is a matrix or a dataframe, then the "original" data are plotted. On the other hand, if expr is a list returned in the 'Estimations' element of TcGSA.LR, then it is those "estimations" made by the TcGSA.LR function that are plotted.

If indiv is 'genes', then each line of the plot is the median of a gene expression over the patients. On the other hand, if indiv is 'patients', then each line of the plot is the median of a patient genes expression in this gene set.

This function uses the Gap statistics to determine the optimal number of clusters in the plotted gene set. See clusGap.

Value

A dataframe the 2 following variables:

- ProbeID which contains the IDs of the probes of the plotted gene set.
- Cluster which to which cluster the probe belongs to.

If clustering is FALSE, then Cluster is NA for all the probes.

Author(s)

Boris P. Hejblum

References

Tibshirani, R., Walther, G. and Hastie, T., 2001, Estimating the number of data clusters via the Gap statistic, *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, **63**, 2: 411–423.

See Also

ggplot, clusGap

Examples

```
## Not run:
data(data_simu_TcGSA)
tcgsa_sim_1grp <- TcGSA.LR(expr=expr_1grp, gmt=gmt_sim, design=design,
                           subject_name="Patient_ID", time_name="TimePoint",
                           time_func="linear", crossedRandom=FALSE)

## End(Not run)

## Not run:
plotSelect.GS(expr=tcgsa_sim_1grp$Estimations, TimePoint=design$TimePoint,
       Subject_ID=design$Patient_ID, gmt=gmt_sim,
```

```
       geneset.names.select=c("Gene set 3", "Gene set 4", "Gene set 5"),
       Subject_ID.select=c("P1", "P2"),
       display="one GS per page",
       time_unit="H",
       lab.cex=0.7
)

## End(Not run)

## Not run:
plotSelect.GS(expr=tcgsa_sim_1grp$Estimations, TimePoint=design$TimePoint,
       Subject_ID=design$Patient_ID, gmt=gmt_sim,
       geneset.names.select=c("Gene set 3", "Gene set 4", "Gene set 5"),
       Subject_ID.select=c("P1", "P2"),
       display="one subject per page",
       time_unit="H",
       lab.cex=0.7
)

## End(Not run)
```

`rmixchisq` *Random Generation of Chi-square Mixtures*

Description

`rmixchisq` is used to simulate a mixture of chi-square distributions that corresponds to the null distribution of the Likelihood Ratio between 2 nested mixed models.

Usage

```
rmixchisq(n, s, q)
```

Arguments

`n`	number of observations.
`s`	number of fixed effects to be tested.
`q`	number of random effects to be tested.

Details

The approximate null distribution of a likelihood ratio for 2 nested mixed models, where both fixed and random effects are tested simultaneously, is a very specific mixture of χ^2 distributions [*Self & Liang (1987), Stram & Lee (1994) and Stram & Lee (1995)*]. It depends on both the number of random effects and the number of fixed effects to be tested simultaneously:

$$LRT_{H_0} \sim \sum_{k=q}^{q+r} \binom{r}{k-q} 2^{-r} \chi^2_{(k)}$$

Value

A vector of random independent observations of the chisquare mixture identified by the values of s and q.

Author(s)

Boris P. Hejblum

References

Self, S. G. and Liang, K., 1987, Asymptotic properties of maximum likelihood estimators and likelihood ratio tests under nonstandard conditions, *Journal of the American Statistical Association* 82: 605–610.

Stram, D. O. and Lee, J. W., 1994, Variance components testing in the longitudinal mixed effects model, *Biometrics* 50: 1171–1177.

Stram, D. O. and Lee, J. W., 1995, Corrections to "Variance components testing in the longitudinal mixed effects model" by Stram, D. O. and Lee, J. W.; 50: 1171–1177 (1994), *Biometrics* 51: 1196.

See Also

pval_simu

Examples

```
library(graphics)
library(stats)

sample_mixt <- rmixchisq(n=1000, s=3, q=3)
plot(density(sample_mixt))
```

---

signifLRT.TcGSA *Identifiying the Significant Gene Sets*

Description

A function that identifies the significant gene sets in an object of class 'TcGSA'.

Usage

```
signifLRT.TcGSA(tcgsa, threshold = 0.05, myproc = "BY",
  nbsimu_pval = 1e+06, write = F, txtfilename = NULL, directory = NULL)
```

Arguments

tcgsa	a tcgsa object.
threshold	the threshold at which the FDR or the FWER should be controlled.
myproc	a vector of character strings containing the names of the multiple testing procedures for which adjusted p-values are to be computed. This vector should include any of the following: "Bonferroni", "Holm", "Hochberg", "SidakSS", "SidakSD", "BH", "BY", "ABH", "TSBH" or "none". "none" indicates no adjustement for multiple testing. See mt.rawp2adjp for details. Default is "BY", the Benjamini & Yekutieli (2001) step-up FDR-controlling procedure (general dependency structures). In order to control the FWER(in case of an analysis that is more a hypothesis confirmation than an exploration of the expression data), we recommand to use "Holm", the Holm (1979) step-down adjusted p-values for strong control of the FWER.
nbsimu_pval	the number of observations under the null distribution to be generated in order to compute the p-values. Default is 1e+06.
write	logical flag enabling the export of the results as a table in a .txt file. Default is FALSE.
txtfilename	a character string with the name of the .txt file in which the results table is to be written, if write is TRUE. Default is NULL.
directory	if write is TRUE, a character string with the directory of the .txt file in which the results table is to be written, if write is TRUE. Default is NULL.

Value

signifLRT.TcGSA returns a list.

The fisrt element mixedLRTadjRes is data frame with p rows (one row for each significant gene set) and the 3 following variables:

- GeneSet the significant gene set name from the gmt object.
- AdjPval the adjusted p-value corresponding to the signicant gene set.
- desc the significant gene set description from the gmt object.

The second element multCorProc passes along the multiple testing procedure used (from the argument myproc).

The third element threshold passes along the significance threshold used (from the argument threshold).

Author(s)

Boris P. Hejblum

References

Hejblum BP, Skinner J, Thiebaut R, (2015) Time-Course Gene Set Analysis for Longitudinal Gene Expression Data. *PLoS Computat Biol* 11(6): e1004310. doi: 10.1371/journal.pcbi.1004310

See Also

multtest.TcGSA, TcGSA.LR

Examples

```
data(data_simu_TcGSA)

tcgsa_sim_1grp <- TcGSA.LR(expr=expr_1grp, gmt=gmt_sim, design=design,
                           subject_name="Patient_ID", time_name="TimePoint",
                           time_func="linear", crossedRandom=FALSE)

sgnifs <- signifLRT.TcGSA(tcgsa_sim_1grp, threshold = 0.05, myproc = "BY",
                          nbsimu_pval = 1000, write=FALSE)
sgnifs
```

---

summary.TcGSA *Summarizing TcGSA*

Description

summary method for class 'TcGSA'

Usage

```
## S3 method for class 'TcGSA'
summary(object, ...)

## S3 method for class 'summary.TcGSA'
print(x, ...)
```

Arguments

object	an object of class 'TcGSA'.
x	an object of class 'summary.TcGSA'.
...	further arguments passed to or from other methods.

Value

The function summary.TcGSA returns a list with the following components (list elements):

- time_func the chosen form for the time trend.
- separateSubjects a logical flag indicating wether gene sets tested for discriminating among patients, or for time trends over time.
- ntg the number of treatment groups.

- ngs the number of tested gene sets.
- nsignif the number of significant gene sets at a 5% FDR (using the default Benjamini & Yekutieli step-up procedure).

Author(s)

Boris P. Hejblum

See Also

TcGSA.LR

Examples

```
data(data_simu_TcGSA)

tcgsa_sim_1grp <- TcGSA.LR(expr=expr_1grp, gmt=gmt_sim, design=design,
                           subject_name="Patient_ID", time_name="TimePoint",
                           time_func="linear", crossedRandom=FALSE)
summary(tcgsa_sim_1grp)

## Not run:
tcgsa_sim_2grp <- TcGSA.LR(expr=expr_2grp, gmt=gmt_sim, design=design,
                           subject_name="Patient_ID", time_name="TimePoint",
                           time_func="linear", crossedRandom=FALSE,
                           group_name="group.var")
summary(tcgsa_sim_2grp)

## End(Not run)
```

TcGSA.LR *Computing the Likelihood Ratios for the Gene Sets under Scrutiny*

Description

This function computes the Likelihood Ratios for the gene sets under scrutiny, as well as estimations of genes dynamics inside those gene sets through mixed models.

Usage

```
TcGSA.LR(expr, gmt, design, subject_name = "Patient_ID",
  time_name = "TimePoint", crossedRandom = FALSE, covariates_fixed = "",
  time_covariates = "", time_func = "linear", group_name = "",
  separateSubjects = FALSE, minGSsize = 10, maxGSsize = 500)

## S3 method for class 'TcGSA'
print(x, ...)
```

Arguments

expr	a matrix or dataframe of gene expression. Its dimension are nxp, with the p samples in column and the n genes in row.
gmt	a **gmt** object containing the gene sets definition. See GSA.read.gmt and definition on www.broadinstitute.org.
design	a matrix or dataframe containing the experimental variables that used in the model, namely subject_name, time_name, and covariates_fixed and time_covariates if applicable. Its dimension are pxm and its row are is in the same order as the columns of expr.
subject_name	the name of the factor variable from design that contains the information on the repetition units used in the mixed model, such as the patient identifiers for instance. Default is 'Patient_ID'. See Details.
time_name	the name of a numeric variable from design that contains the information on the time replicates (the time points at which gene expression was measured). Default is 'TimePoint'. See Details.
crossedRandom	logical flag indicating wether the random effects of the subjects and of the time points should be modeled as one crossed random effect or as two separated random effects. Default is FALSE. See details.
covariates_fixed	a character vector with the names of numeric or factor variables from the design matrix that should appear as fixed effects in the model. See details. Default is "", which corresponds to no covariates in the model.
time_covariates	a character vector with the names of numeric or factor variables from the design matrix that should appear as fixed effects interaction with the time_name variable in the model. See details. Default is "", which corresponds to no covariates in the model.
time_func	the form of the time trend. Can be either one of "linear", "cubic", "splines" or specified by the user, or the column name of a factor variable from design. If specified by the user, it must be as an expression using only names of variables from the design matrix with only the three following operators: +, *, / . The "splines" form corresponds to the natural cubic B-splines (see also ns). If there are only a few timepoints, a "linear" form should be sufficient. Otherwise, the "cubic" form is more parsimonious than the "splines" form, and should be sufficiently flexible. If the column name of a factor variable from design is supplied, then time is considered as discrete in the analysis. If the user specify a formula using column names from design, both factor and numeric variables can be used.
group_name	in the case of several treatment groups, the name of a factor variable from the design matrix. It indicates to which treatment group each sample belongs to. Default is "", which means that there is only one treatment group. See Details.
separateSubjects	logical flag indicating that the analysis identifies gene sets that discriminates patients rather than gene sets than have a significant trend over time. Default is FALSE. See Details.

minGSsize	the minimum number of genes in a gene set. If there are less genes than this number in one of the gene sets under scrutinity, the Likelihood Ratio of this gene set is not computed (the mixed model are not fitted). Default is 10 genes as the minimum.
maxGSsize	the maximum number of genes in a gene set. If there are more genes than this number in one of the gene sets under scrutinity, the Likelihood Ratio of this gene set is not computed (the mixed model are not fitted). This is to avoid very long computation times. Default is 500 genes as the maximum.
x	an object of class 'TcGSA'.
...	further arguments passed to or from other methods.

Details

This Time-course Gene Set Analysis aims at identifying gene sets that are not stable over time, either homogeneously or heterogeneously (see *Hejblum et al, 2012*)in terms of their probes. And when the argument separateSubjects is TRUE, instead of identifying gene sets that have a significant trend over time, *TcGSA* identifies gene sets that have significantly different trends over time depending on Subjects.

Value

TcGSA.LR returns a tcgsa object, which is a list with the 5 following elements:

- fit a data frame that contains the 3 following variables:
  - LR: the likelihood ratio between the model under the null hypothesis and the model under the alternative hypothesis.
  - CVG_H0: convergence status of the model under the null hypothesis.
  - CVG_H1: convergence status of the model under the alternative hypothesis.
- time_func: a character string passing along the value of the time_func argument used in the call.
- GeneSets_gmt: a gmt object passing along the value of the gmt argument used in the call.
- group.var: a factor passing along the group_name variable from the design matrix.
- separateSubjects: a logical flag passing along the value of the separateSubjects argument used in the call.
- Estimations: a list of 3 dimensions arrays. Each element of the list (i.e. each array) corresponds to the estimations of gene expression dynamics for each of the gene sets under scrutiny (obtained from mixed models). The first dimension of those arrays is the genes included in the concerned gene set, the second dimension is the Patient_ID, and the third dimension is the TimePoint. The values inside those arrays are estimated gene expressions.
- time_DF: the degree of freedom of the natural splines functions

Author(s)

Boris P. Hejblum

References

Hejblum BP, Skinner J, Thiebaut R, (2015) Time-Course Gene Set Analysis for Longitudinal Gene Expression Data. *PLoS Computat Biol* 11(6): e1004310. doi: 10.1371/journal.pcbi.1004310

See Also

summary.TcGSA, plot.TcGSA, and TcGSA.LR.parallel for an implementation using parallel computing

Examples

```
data(data_simu_TcGSA)

tcgsa_sim_1grp <- TcGSA.LR(expr=expr_1grp, gmt=gmt_sim, design=design,
                           subject_name="Patient_ID", time_name="TimePoint",
                           time_func="linear", crossedRandom=FALSE)
tcgsa_sim_1grp
summary(tcgsa_sim_1grp)


## Not run:
plot(x=tcgsa_sim_1grp, expr=expr_1grp,
    Subject_ID=design$Patient_ID, TimePoint=design$TimePoint,
    baseline=1,
    B=100,
    time_unit="H"
    )

## End(Not run)

## Not run:
tcgsa_sim_2grp <- TcGSA.LR(expr=expr_2grp, gmt=gmt_sim, design=design,
                           subject_name="Patient_ID", time_name="TimePoint",
                           time_func="linear", crossedRandom=FALSE,
                           group_name="group.var")
tcgsa_sim_2grp


## End(Not run)
```

TcGSA.LR.parallel *Parallel computing the Likelihood Ratios for the Gene Sets under Scrutiny*

Description

A parallel version of the function TcGSA.LR to be used on a cluster of computing processors. This function computes the Likelihood Ratios for the gene sets under scrutiny, as well as estimations of genes dynamics inside those gene sets through mixed models.

Usage

```
TcGSA.LR.parallel(Ncpus, type_connec, expr, gmt, design,
  subject_name = "Patient_ID", time_name = "TimePoint",
  crossedRandom = FALSE, covariates_fixed = "", time_covariates = "",
  time_func = "linear", group_name = "", separateSubjects = FALSE,
  minGSsize = 10, maxGSsize = 500, monitorfile = "")
```

Arguments

Ncpus	The number of processors available on the cluster.
type_connec	The type of connection between the processors. Supported cluster types are "SOCK", "PVM", "MPI", and "NWS". See also makeCluster.
expr	a matrix or dataframe of gene expression. Its dimension are nxp, with the p samples in column and the n genes in row.
gmt	a **gmt** object containing the gene sets definition. See GSA.read.gmt and definition on www.broadinstitute.org.
design	a matrix or dataframe containing the experimental variables that used in the model, namely subject_name, time_name, and covariates_fixed and time_covariates if applicable. Its dimension are pxm and its row are is in the same order as the columns of expr.
subject_name	the name of the factor variable from design that contains the information on the repetition units used in the mixed model, such as the patient identifiers for instance. Default is 'Patient_ID'. See Details.
time_name	the name of the numeric or factor variable from design contains the information on the time replicates (the time points at which gene expression was measured). Default is 'TimePoint'. See Details.
crossedRandom	logical flag indicating wether the random effects of the subjects and of the time points should be modeled as one crossed random effect or as two separated random effects. Default is FALSE. See details.
covariates_fixed	a character vector with the names of numeric or factor variables from the design matrix that should appear as fixed effects in the model. See details. Default is "", which corresponds to no covariates in the model.
time_covariates	the name of a numeric variable from design that contains the information on the time replicates (the time points at which gene expression was measured). Default is 'TimePoint'. See Details.
time_func	the form of the time trend. Can be either one of "linear", "cubic", "splines" or specified by the user, or the column name of a factor variable from design. If specified by the user, it must be as an expression using only names of variables from the design matrix with only the three following operators: +, *, / . The "splines" form corresponds to the natural cubic B-splines (see also ns). If there are only a few timepoints, a "linear" form should be sufficient. Otherwise, the "cubic" form is more parsimonious than the "splines" form, and should be sufficiently flexible. If the column name of a factor variable from

design is supplied, then time is considered as discrete in the analysis. If the user specify a formula using column names from design, both factor and numeric variables can be used.

group_name in the case of several treatment groups, the name of a factor variable from the design matrix. It indicates to which treatment group each sample belongs to. Default is "", which means that there is only one treatment group. See Details.

separateSubjects logical flag indicating that the analysis identifies gene sets that discriminates patients rather than gene sets than have a significant trend over time. Default is FALSE. See Details.

minGSsize the minimum number of genes in a gene set. If there are less genes than this number in one of the gene sets under scrutinity, the Likelihood Ratio of this gene set is not computed (the mixed model are not fitted). Default is 10 genes as the minimum.

maxGSsize the maximum number of genes in a gene set. If there are more genes than this number in one of the gene sets under scrutinity, the Likelihood Ratio of this gene set is not computed (the mixed model are not fitted). This is to avoid very long computation times. Default is 500 genes as the maximum.

monitorfile a writable connections or a character string naming a file to write into, to monitor the progress of the analysis. Default is "" which is no monitoring. See Details.

Details

This Time-course Gene Set Analysis aims at identifying gene sets that are not stable over time, either homogeneously or heterogeneously (see *Hejblum et al, 2012*) in terms of their probes. And when the argument separatePatients is TRUE, instead of identifying gene sets that have a significant trend over time (possibly with probes heterogeneity of this trend), *TcGSA* identifies gene sets that have significantly different trends over time depending on the patient.

If the monitorfile argument is a character string naming a file to write into, in the case of a new file that does not exist yet, such a new file will be created. A line is written each time one of the gene sets under scrutiny has been analysed (i.e. the two mixed models have been fitted, see TcGSA.LR) by one of the parallelized processors.

Value

TcGSA.LR returns a tcgsa object, which is a list with the 5 following elements:

- fit a data frame that contains the 3 following variables:
  - LR: the likelihood ratio between the model under the null hypothesis and the model under the alternative hypothesis.
  - CVG_H0: convergence status of the model under the null hypothesis.
  - CVG_H1: convergence status of the model under the alternative hypothesis.
- time_func: a character string passing along the value of the time_func argument used in the call.
- GeneSets_gmt: a gmt object passing along the value of the gmt argument used in the call.
- group.var: a factor passing along the group_name variable from the design matrix.

- `separateSubjects`: a logical flag passing along the value of the `separateSubjects` argument used in the call.
- `Estimations`: a list of 3 dimensions arrays. Each element of the list (i.e. each array) corresponds to the estimations of gene expression dynamics for each of the gene sets under scrutiny (obtained from mixed models). The first dimension of those arrays is the genes included in the concerned gene set, the second dimension is the `Patient_ID`, and the third dimension is the `TimePoint`. The values inside those arrays are estimated gene expressions.
- `time_DF`: the degree of freedom of the natural splines functions

Author(s)

Boris P. Hejblum

References

Hejblum BP, Skinner J, Thiebaut R, (2015) Time-Course Gene Set Analysis for Longitudinal Gene Expression Data. *PLoS Computat Biol* 11(6): e1004310. doi: 10.1371/journal.pcbi.1004310

See Also

`summary.TcGSA`, `plot.TcGSA`

Examples

```
data(data_simu_TcGSA)

tcgsa_sim_1grp <- TcGSA.LR(expr=expr_1grp, gmt=gmt_sim, design=design,
                           subject_name="Patient_ID", time_name="TimePoint",
                           time_func="linear", crossedRandom=FALSE)

## Not run:
require(doParallel)
tcgsa_sim_1grp <- TcGSA.LR.parallel(Ncpus = 2, type_connec = 'SOCK',
                             expr=expr_1grp, gmt=gmt_sim, design=design,
                             subject_name="Patient_ID", time_name="TimePoint",
                             time_func="linear", crossedRandom=FALSE,
                             separateSubjects=TRUE)

## End(Not run)
tcgsa_sim_1grp
summary(tcgsa_sim_1grp)
```

Index